Brady McAtee

bradymcatee12@gmail.com | bradymcatee.com | github.com/bradymcatee | linkedin.com/in/brady-mcatee

Education

Portland State University, BS in Computer Science, Minor in Physics — Sept 2023 – March 2026

- GPA: 3.9/4.0
- **Coursework:** Data Structures and Algorithms, Operating Systems, Computer Graphics, Computer Vision

Experience

Software Engineer Intern, MSEI Biotronik – Lake Oswego, OR — June 2025 – Present

- Built simulation models in C#/.NET for multiple hardware components, enabling R&D engineers to test new device designs entirely in software
- Ported a legacy .NET Framework WPF library to .NET 8, preserving the API and ensuring compatibility across modern and legacy applications
- Refactored a graphical hardware simulation tool by separating core logic from UI for cleaner architecture and easier maintenance
- Worked in an Agile environment using Git and Azure DevOps Server for version control and collaboration across the software and hardware teams

Projects

recipeBox — github.com/bradymcatee/recipeBox

- Built a full-stack web app for restaurants to manage recipes and ingredients with full CRUD support
- Designed RESTful APIs and normalized PostgreSQL schemas to keep recipe and ingredient data consistent
- Implemented responsive React UI with intuitive forms for adding, editing, and searching recipes
- Focused on clean architecture and efficient data handling for a smooth, real-world user experience
- Deployed app on AWS EC2 for restaurant staff use

C++ Ray Tracer — github.com/bradymcatee/RayTracer

- Built a C++ CPU ray tracer that renders 3D scenes with reflections, refractions, anti-aliasing, and depth of field
- Used OOP principles to structure the engine into small, reusable components with a CLI to tune quality
- Parallelized with OpenMP and a thread-safe RNG, reducing render times by 3x on my 8-core CPU
- Implemented physically based materials (Lambertian, metal, glass) and reliable hit math for clean images

Tennis Shot Tracker — github.com/bradymcatee/servetracker

- Built TrackNet-style PyTorch model (3-frame, 9-ch) for tennis ball heatmap detection with focal BCE + Adadelta
- Implemented dataset loader and training loop with checkpoints, custom accuracy metric, and training curves
- Shipped OpenCV inference CLI: preprocessing, RANSAC+LMEDS homography, trajectory overlays
- Computed real-world speeds via homography, per-frame velocity, Savitzky–Golay smoothing, and mph statistics

Interactive Photo Refocusing App — github.com/bradymcatee/refocusing

- Built full-stack app using Python, PyTorch, and Flask for DSLR-style depth-of-field simulation from single images
- Implemented Multi-Scale CNN (87.9M parameters) achieving 79% accuracy for monocular depth estimation
- Created responsive web interface with real-time click-to-focus interaction and configurable aperture effect
- Deployed production-ready solution with Docker and cloud deployment options

Technologies

Languages: C++, C, C#, Python, Java, JavaScript, SQL

Technologies: .NET, Node.js, Express, PostgreSQL, Docker, Azure DevOps, Git, PyTorch, Jest, PyTest, NUnit, Linux